[0184] An Entity Attribute is an Attribute that is stored as a column in an Entity Attribute table that is related to the parent Entity table using the Entity ID. An Entity Attribute may include historical values, but that is not necessary, and an Entity Attribute can be either single- or multi-valued. The following describes the preferred method for including an Entity Attribute table in a query, assuming the parent Entity table is included in the query:

[0185] 1) Add the column name for the Entity Attribute to the `SELECT` clause.

[0186] 2) Add the table name for the Attribute table to the `FROM` clause.

[0187] 3) Add the following Entity Attribute relationship to the `WHERE` clause

```
AND ([parent Entity table].[Entity ID name] = [Entity
Attribute table name].[Entity ID name]).
```

[0188] 4) For a historical query of a historical Entity Attribute, add the following historical relationship to the `WHERE` clause

```
AND ([Entity Attribute table name].Begin_Dte <
[historical date])
AND ([historical date] <= NVL([Entity Attribute table
name].End_Dte,SYSDATE).
```

[0189] 5) For a non-historical query of a historical Entity Attribute, add the following historical relationship to the `WHERE` clause

```
AND ([Entity Attribute table name].End_Dte IS NULL).
```

[0190] A dynseg Attribute is an Attribute that is stored as a column in a dynseg Attribute table that is related to the parent Entity table using the Entity ID, `Begin_Pct`, and `End_Pct`. The parent Entity of a dynseg Attribute must be an Entity with a linear geometry. A dynseg Attribute may include historical values, but that is not necessary, and can be either single- or multi-valued. The following is the preferred method for including a dynseg Attribute table in a query, assuming the parent Entity table is included in the query. Because the join required depends on the number of dynseg tables included, the instructions demonstrate how to join two dynseg Attribute tables and describe how to extend this to more than two tables.

[0191] 1) Add the column name for the dynseg Attributes to the `SELECT` clause.

[0192] 2) Add the table names for the dynseg Attribute tables to the `FROM` clause.

[0193] 3) To include the dynseg percents in the query results, add the following to the `SELECT` clause

```
GREATEST([dynseg Attribute table name 1].Begin_Pct,
[dynseg Attribute table name 2].Begin_Pct) Begin_Pct,
LEAST([dynseg Attribute table name 1].End_Pct, [dynseg
Attribute table name 2].End_Pct) End_Pct.
```

[0194] To include additional dynseg Attribute tables, add the `Begin_Pct` of each table to the list of columns in the `GREATEST` function and add the `End_Pct` of each table to the list of columns in the `LEAST` function. If only one dynseg Attribute table is included in the join, the `GREATEST` and `LEAST` functions are not required. If other dynseg joins are involved in a query, the `GREATEST` and `LEAST` statements

must be modified to include the Begin_Pct and End_Pct columns to support these joins.

[0195] 4) Add the following dynseg Attribute relationship to the WHERE clause

```
AND ([parent Entity table].[Entity ID name] = [dynseg
Attribute table name].[Entity ID name]
AND (GREATEST([dynseg Attribute table name 1].Begin_Pct,
[dynseg Attribute table name 2].Begin_Pct)
< LEAST([dynseg Attribute table name
1].End_Pct, [dynseg Attribute table name 2].End_Pct))
```

[0196] To include additional dynseg Attribute tables, add the Begin_Pct of each table to the list of columns in the GREATEST function and add the End_Pct of each table to the list of columns in the LEAST function. If only one dynseg Attribute table is included in the join, the second half of the relationship (i.e., the relationship associated with the second AND) is not required. If other dynseg joins are involved in a query, the GREATEST and LEAST statements must be modified to include the Begin_Pct and End_Pct columns to support these joins.

[0197] 5) For a historical query of a historical Entity Attribute, add the following historical relationship to the WHERE clause, one such statement for each historical dynseg Attribute table:

```
AND ([dynseg Attribute table name].Begin_Dte <
[historical date])
```